

Cahier des charges techniques

Distributech ETL

Sommaire

1-Vision produit.....	1
2-User Stories	2
3-Rédaction itérative	3
4-Spécifications techniques.....	3
5-KPI & critères d'acceptation	4
6-Organisation agile du projet	5
7-Points de vigilance IA	6

1-Vision produit

Objectif : Cadrer rapidement le sens du projet IA

Contenu :

Pour pallier le manque de suivi logistique fiable et en temps réel, un pipeline ETL a été mis en place pour intégrer les données de commandes (CSV) et de stocks (SQLite) dans une base MySQL centralisée. Celle-ci permet de suivre l'historique des commandes, visualiser les stocks par région et piloter efficacement les opérations commerciales et logistiques.

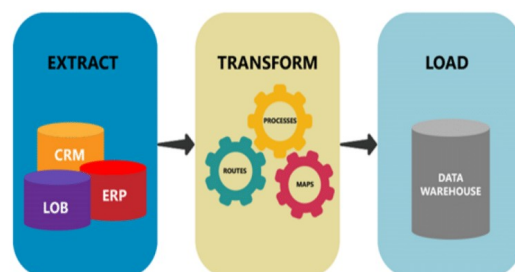
Cette architecture simple, robuste, évolutive et maîtrisable par les équipes en place permet de fiabiliser rapidement les données tout en posant une base solide pour les usages futurs.

Checklist / Questions clés :

1. Quel problème métier cherchons-nous à résoudre ?
2. Quelle valeur pour l'utilisateur final ?
3. Pourquoi une IA plutôt qu'une autre solution ?

Agile :

1. Résumée en une phrase, revue à chaque grand jalon produit



2-User Stories

Objectif : Traduire les besoins en fonctionnalités concrètes

Exemples :

User Story no. 1 En tant que *commercial terrain*, je veux un *fichier commande nettoyé* afin d'*améliorer le traitement du process de commande*.

Critères d'acceptation mesurables :

1. Le fichier ne contient aucune ligne vide ni doublon.
2. Les formats de date sont homogènes (ex. AAAA-MM-JJ).
3. n>95 % des champs obligatoires sont remplis (client, produit, quantité).

User Story no. 2 En tant que *responsable commercial*, je veux *recupérer un rapport hebdomadaire des stocks*** afin d'*avoir une vision à jour des tendances et tensions logistiques*.

Critères d'acceptation mesurables :

1. Le rapport hebdomadaire est généré automatiquement.
2. Il contient les stocks par produit et par région à la date de génération.
3. Le fichier au format csv est accessible dans un dossier.

User Story no. 3 En tant que *responsable commercial*, je veux *intégrer les commandes et les productions dans un stock centralisé* afin de *garder à jour des stocks cohérents et piloter efficacement les opérations commerciales*.

Critères d'acceptation mesurables :

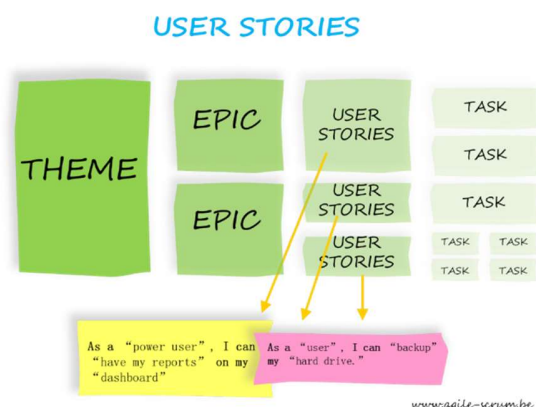
1. n>95 % des commandes validées sont intégrées au stock central.
2. Toutes les entrées et sorties de stock sont historisées avec date, origine et quantité.
3. Les stocks sont cohérents entre les différentes sources (aucun écart supérieur à 1 %).
4. Les données sont accessibles dans un fichier csv et mise à jour est hebdomadaire.

Checklist / Questions clés :

1. En tant que... je veux... afin de... ?
2. Quels bénéfices utilisateurs ?
3. Critères d'acceptation mesurables définis ?

Agile :

1. Prioriser avec le Product Owner à chaque sprint



3-Rédaction itérative

Objectif : Permettre l'évolution naturelle du projet

Contenu :

Révision continue des besoins à chaque sprint : À l'issue de chaque sprint, les exigences fonctionnelles et techniques sont réévaluées en tenant compte des retours du terrain, des échanges avec les commerciaux et des données collectées. Cette démarche garantit une adaptation progressive du système à son environnement réel d'exploitation.

Intégration des apprentissages POC, MVP et tests terrain : Les enseignements tirés du POC (validation des sources et du pipeline), du MVP (structure minimale fonctionnelle de la base et du script ETL), ainsi que des tests terrain (exploitabilité par les commerciaux et fiabilité des données de stock) sont systématiquement intégrés aux itérations suivantes. Ces retours permettent d'optimiser la robustesse du système, d'affiner la logique métier et de renforcer la maintenabilité de la solution.

Checklist / Questions clés :

1. Revoir les besoins et contraintes à chaque sprint
2. Inclure apprentissages du POC, MVP, tests terrain

Agile :

1. Mises à jour fréquentes, versionnées

4-Spécifications techniques

Objectif : Poser les bases technologiques sans figer

Contenu :

1. Python 3.12, MySQL 8, Pandas 2.3
2. Connexion MySQL via mysql-connector 9.3
3. Docker + Adminer pour BDD
4. Tests Pytest(v8.4) + Pytest-cov(v6.2)
5. Données d'entrée : CSV & SQLite



Checklist / Questions clés :

1. Langage / framework IA défini
2. Données nécessaires (type, volume, accès, RGPD)
3. Contraintes techniques (perfs, compatibilité, stockage)

Agile :

1. Ajouter les specs par briques fonctionnelles au fil des itérations

5-KPI & critères d'acceptation

Objectif : Valider la qualité et l'utilité des livrables

Exemples de KPI :

User Story no. 1 KPIs:

1. Taux de doublons : 2 % (aucune ligne en double dans le fichier).
2. Taux de lignes vides : 2 % (chaque ligne contient des données utiles).
3. Taux de complétude : 98 % des lignes doivent avoir les champs obligatoires renseignés (client, produit, quantité).
4. Taux de formats conformes (dates) : 98 % des dates au format AAAA-MM-JJ.

User Story no. 2 KPIs:

1. Taux d'erreurs d'accès : 2 % (le fichier CSV est bien généré et accessible dans le bon dossier).
2. Taux de complétude du rapport : tous les produits et toutes les régions doivent être représentés.
3. Temps de génération du rapport : < 5 seconde.
4. Format de fichier conforme : CSV bien structuré (lisible, séparateur correct, encodage UTF-8).

User Story no. 3 KPIs:

1. Taux d'intégration des commandes validées : 98 %.
2. Taux de mouvements historisés : 98 % des entrées/sorties doivent être enregistrées avec date, origine, quantité.
3. Écart de cohérence entre sources : ≤ 1 % entre le stock théorique et les données d'entrée (commandes, production).
4. Fichier export disponible : chaque mise à jour génère un fichier CSV complet et lisible.
5. **Checklist / Questions clés :**
6. Précision / rappel / F1-score
7. Temps de réponse acceptable
8. Taux de faux positifs / négatifs

Agile :

1. Définis pour chaque User Story, évalués en fin de sprint



6-Organisation agile du projet

Objectif : Structurer l'avancement en mode collaboratif

Contenu :

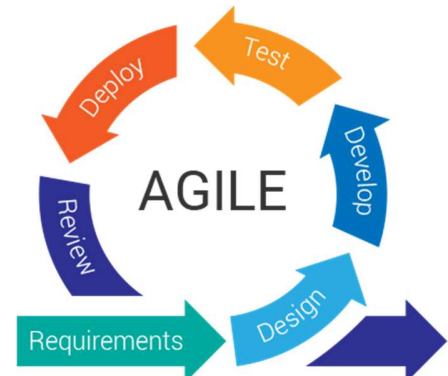
1. 3 sprints courts (1 semaine)
2. Rituels : daily, sprint planning, review, rétrospective
3. Suivi via Notion (retroplanning, tickets, documentation, RGPD)

Checklist / Questions clés :

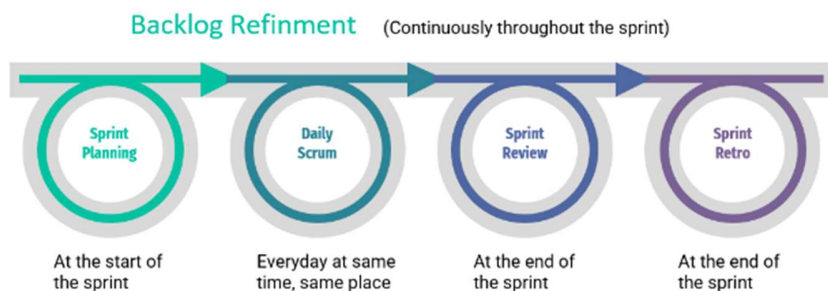
1. Sprints courts définis (1-3 semaines)
2. Rituels agiles respectés (planning, daily, review, rétro)
3. Suivi backlog (Jira, Trello, Notion...)

Agile :

1. Responsabiliser l'équipe, suivre l'avancement de façon visuelle



Sprint Ceremonies



7-Points de vigilance IA

Objectif : Anticiper les pièges spécifiques aux projets IA



Risques & Préventions :

- Sur-nettoyage des données
 - Risque : suppression involontaire de données valides jugées ""anormales"", insertion de données erronées dans la BDD centralisée
 - Conséquence : perte d'informations utiles, résultats faussés.
 - Prévention : valider les règles de nettoyage avec les utilisateurs métier.
- Données biaisées
 - Risque : les données reflètent des pratiques locales ou des périodes spécifiques, ""Boite noire"" règles de transformation non-explicables
 - Conséquence : analyses non généralisables, décisions faussées.
 - Prévention : vérifier la représentativité des données (produits, régions, revendeurs, etc.).
- Manque d'explicabilité
 - Risque : traitements ou règles métier opaques, Manque de log des traitements
 - Conséquence : incompréhension, difficulté à corriger ou justifier les résultats.
 - Prévention : documenter clairement chaque transformation, privilégier des règles simples et traçables.
- Absence de versioning
 - Risque : perte de maîtrise sur les versions de code ou de données.
 - Conséquence : résultats incohérents, erreurs non reproductibles.
 - Prévention : versionner le code (Git), horodater les fichiers sources, documenter les évolutions de schéma.
- Tests insuffisants en conditions réelles
 - Risque : pipeline testé uniquement sur des cas simplifiés.
 - Conséquence : erreurs silencieuses en production, instabilité.
 - Prévention : tester sur des données réelles, puis sur des volumes complets avant mise en production.

Checklist / Questions clés :

1. Données biaisées identifiées ?
2. Explicabilité du modèle assurée ?
3. Versioning ?
4. Tests sur données réelles ?

Agile :

1. Maintenir checklist IA spécifique à jour à chaque étape